NAIST Simultaneous Speech Translation System for IWSLT 2025

Haotian Tan[1], Ruhiyah Faradishi Widiaputri[1], Jan Meyer Saragih[1], Yuka Ko[1], Katsuhito Sudoh[1,2], Satoshi Nakamura[1,3], Sakriani Sakti[1],
[1]Nara Institute of Science and Technology, Japan,
[2]Nara Women's University, Japan,
[3]Chinese University of Hong Kong, Shenzhen, China,
Correspondence: tan.haotian.tf5@naist.ac.jp, ssakti@is.naist.jp

Abstract

This paper describes the NAIST submission to the English-to-{German, Japanese, Chinese} Simultaneous Speech-to-Text track at IWSLT 2025. Last year, our system was based on an end-to-end speech-to-text translation model that combined HuBERT and mBART. This year, the system consists of a Whisper encoder, the DeCo compressive projector, and the Qwen large language model. The simultaneous translation (SimulST) system is implemented by applying a local agreement policy to an offline-trained translation model. For the streaming translation (StreamST) system, we integrate an online version of the SHAS segmenter into our SimulST architecture. Our results demonstrate that adopting LLMs as the backbone architecture for speech translation tasks yields strong translation performance. Additionally, leveraging robust segmentation capability of SHAS for StreamST achieves good quality-latency trade-off when processing unbounded audio streams.

1 Introduction

Simultaneous speech-to-text translation (SimulST) aims to mimic human interpreters by providing real-time translation with low latency while maintaining high translation quality. In SimulST, the system generates translation before receiving the full source utterance. A decision policy is required to determine whether to generate partial output or wait for additional source context to improve reliability.

Some prior studies train dedicated models for SimulST using specialized training strategies and architecture designs to learn a data-driven decision policy (Ma et al., 2020b; Ren et al., 2020; Zeng et al., 2021; Liu et al., 2021; Zhang et al., 2024). However, their performance heavily depends on the design of training strategies, which is a complex and challenging task. Furthermore, achieving different latency regimes typically requires training multiple separate models, substantially increasing computational requirements and complicating practical deployment.

Due to the aforementioned reasons, approaches using a single model for different simultaneous scenarios have become popular (Papi et al., 2022a). These methods train the speech translation (ST) model using offline translation data and then apply a manually designed decision policy to this offline ST model for SimulST inference. In this way, a single ST model can adapt to different latency requirements in practical use. Designing an optimal decision policy is significant to their performance. Among several existing decision policies (Ma et al., 2019; Liu et al., 2020; Nguyen et al., 2021), Local Agreement (LA) (Liu et al., 2020; Polák et al., 2022) is one of the most popular method and won the SimulST track of IWSLT 2022 (Polák et al., 2022). It makes decisions by establishing an agreement between two consecutive chunks and only emitting their longest common prefixes. Additionally, the attention-based decision policies, EDAtt (Papi et al., 2023a) and AlignAtt (Papi et al., 2023b), have been proposed for encoder-decoder ST models. They leverage the cross-attention mechanism to make decisions based on the idea that if the model attends to the tail end of the incomplete input speech, the generated hypothesis is unreliable and more context is needed. These attention-based decision policies have shown good performance and have been widely adopted for SimulST tasks (Ko et al., 2024; Tan and Sakti, 2024).

Most recently, several studies have explored the use of pre-trained large language models (LLMs) for SimulST, capitalizing on their powerful generative and zero-shot transfer capabilities. Koshkin et al. (2024) proposes a cascaded architecture combining an ASR model with a decoder-only LLM to perform SimulST. However, this cascaded approach is hindered by error propagation and additional latency. A few works have instead focused

Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025), pages 369–378
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

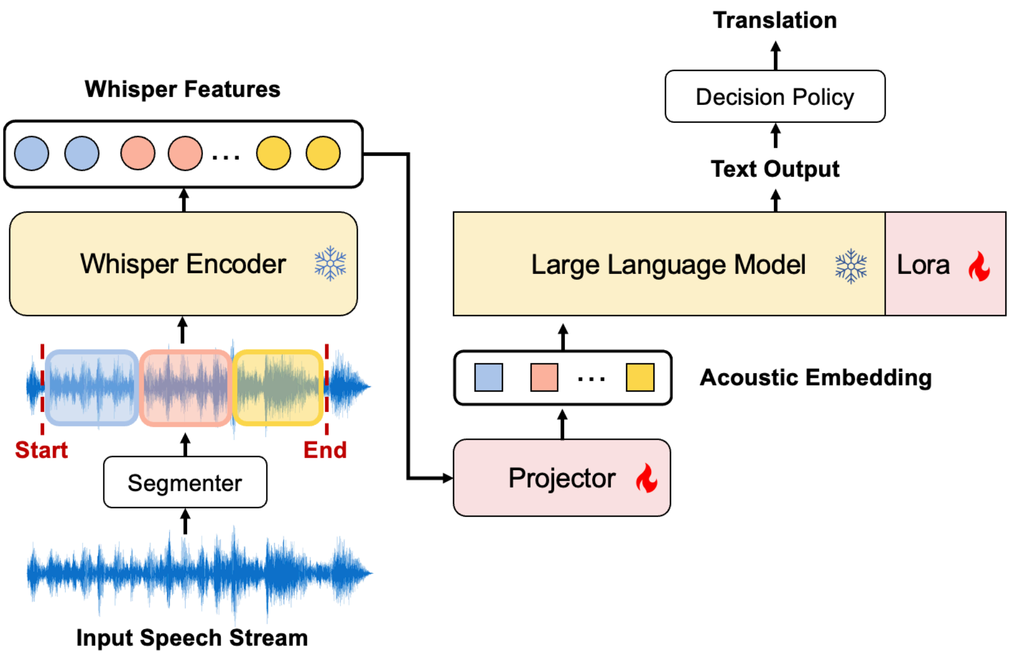


Figure 1: Architecture of our LLM-based StreamST system. The model integrates a Whisper encoder with the LLM via the projector module. The decision policy enables simultaneous translation capabilities, while an online segmenter processes unbounded audio streams for real-time streaming translation.

on end-to-end LLM-based SimulST systems. Xu et al. (2024) trains an offline LLM-based ST model and extends it to SimulST using the Hold-n (Liu et al., 2020) decision policy. Fu et al. (2025) develops a fully end-to-end system through a specialized multi-step training strategy. Another line of work by Ouyang et al. (2025) reformulates SimulST as a multi-turn dialogue task, enabling the LLM to make translation decisions by predicting an end-of-turn token.

Nevertheless, most of the aforementioned SimulST systems are designed to work on pre-segmented speech. Streaming speech-to-text translation (StreamST), the task of automatically translating speech while incrementally receiving an audio stream, remains a challenging problem due to the need for effectively processing the history audio and text contexts. Papi et al. (2024) introduces the first StreamST policy to deal with the unbounded audio stream via audio and textual history selection. Ouyang et al. (2025) utilizes a LLM cache management module to handle the unbounded audio stream during inference.

This paper describes the NAIST submission for the English-to-{German, Japanese, Chinese} Simultaneous Speech-to-Text Track at IWSLT 2025. In our last year's system (Ko et al., 2024), we applied the LA policy to an encoder-decoder model to do SimulST. For the IWSLT 2025 Evaluation Campaign, we explore employing LLM in our system to conduct translation in real time. We construct an end-to-end LLM-based ST model, trained on offline data, and—similar to our previous system—enable it to perform simultaneous translation using the LA policy. To handle the unbounded audio stream in real-world settings, we adopt an online version of the SHAS segmentation method (Tsiamas et al., 2022) to identify the speech segments in the audio stream and present the SHAS-based StreamST.

2 System Description

In this section, we first describe the model architecture of our system and its training methodology. Then we present the detailed implementation of our simultaneous speech-to-text translation and streaming speech-to-text translation approaches.

2.1 Model Architecture

As illustrated in Figure 1, the translation model of our system comprises three principal components: a Whisper encoder, a projector, and a large language model. The Whisper output features of the input speech are transformed into acoustic embeddings, which are subsequently integrated with the prompt textual embeddings and fed into the LLM to generate the target translation.

Whisper Encoder: The Whisper model (Radford et al., 2023) is an open-source speech model trained on a large amount of speech recognition and translation data. The output features of the Whisper encoder have demonstrated superior performance in modeling speech information and have been widely adopted for downstream speech processing tasks. In our submission system, we utilize the Whisper-large-v3[1] architecture to extract high-fidelity acoustic features from the source speech signal.

Projector: The projector serves as a critical bridging mechanism to address the speech-text modality gap between the source speech and the text-driven LLM by mapping the acoustic features into the LLM embedding space. In our system, we implement DeCo (Yao et al., 2024) as the projector between the Whisper encoder and the LLM. DeCo is a compressive projector originally proposed for visual-language models that exhibits a remarkably efficient structure: a 2D adaptive averaging pooling (AdaptiveAvgPool) layer functioning as a down-sampler, followed by two linear projection layers. These linear projection layers constitute the only trainable parameters in this module, making it computationally efficient while effectively aligning the speech representations with the LLM embedding space.

Large Language Model: The Qwen-2.5-7B LLM[2] (Yang et al., 2024) is employed in our system to function as an expert translator. The model processes the acoustic embeddings alongside textual prompts to generate high-quality translations based on the prompt instruction. The generative capabilities of the LLM enable flexible adaptation to various translation scenarios while maintaining semantic accuracy and linguistic fluency in the target language.

2.2 Model Training

2.2.1 Training Objective

We train our system in an offline manner using supervised learning with parallel speech-text data. Specifically, given the training dataset $D = \{(\mathbf{S}, \mathbf{Y}_{src}, \mathbf{Y}_{tgt})\}$, the Whisper encoder $\mathcal{F}_e(\cdot)$ consumes the complete source speech signal $\mathbf{S} = \{s_1, s_2, ..., s_T\}$ to extract acoustic features:

$$\mathbf{X_s} = \mathcal{F}_e(\mathbf{S}) = \{x_1, x_2, ...x_L\}. \quad (1)$$

The projector $\mathcal{F}_p(\cdot)$ subsequently maps these acoustic features into the LLM embedding space with length compression to generate the acoustic embedding of the source speech:

$$\mathbf{E}(\mathbf{X_s}) = \mathcal{F}_p(\mathbf{X_s}) = \{e_1, e_2, ..., e_M\}, M < L. \quad (2)$$

We integrate the acoustic embedding $\mathbf{E}(\mathbf{X_s})$ with the textual embedding of the LLM prompt and the prefix tokens to form the composite input for the LLM:

$$\mathbf{I}_{llm} = \{\mathbf{E}(\mathbf{X_s}), \mathbf{E}(Prompt), \mathbf{E}(Prefix)\}. \quad (3)$$

The LLM then processes this multimodal input to autoregressively get the model output:

$$P(\mathbf{Y}|\mathbf{I}_{llm}) = \mathcal{F}_{llm}(\mathbf{I}_{llm}), \quad (4)$$

where $\mathbf{Y} = \{y_1, y_2, ..., y_N\}$ denotes the target textual sequence during training. Given the composite LLM input $\mathbf{I}_{llm}$, we optimize the system by minimizing the token-level negative log-likelihood loss over the target output sequence:

$$\mathcal{L} = -\frac{1}{|\mathbf{Y}|}\sum_{i=1}^{|\mathbf{Y}|}\log P(y_i|\mathbf{I}_{llm}, y_{<i}). \quad (5)$$

2.2.2 ASR Joint Training

To enhance the performance of the translation system and facilitate training, we implement a multi-task learning approach utilizing automatic speech recognition (ASR) as an auxiliary task. Unlike approaches proposed by Chen et al. (2024) and Huang et al. (2024), which employ a dedicated prompt for the transcription task to augment the training data, we utilizes a single unified prompt that instructs the LLM to generate the transcription immediately following its translation output. The target sequence for training is specifically formatted as:

$\mathbf{Y}$ = Translation:$\mathbf{Y}_{tgt}$ <end> Transcription:$\mathbf{Y}_{src}$,

where the <end> token denotes the end of the translation, which is a signal to terminate the decoding process during inference when only the translation component is required for deployment scenarios.

2.2.3 Fine-tuning

During the training phase, the pretrained weights of both the whisper encoder and the core LLM architecture are frozen to maintain their representational capabilities. We fine-tune the LLM using Low-Rank Adaptation (LoRA) (Hu et al., 2022) and optimize the complete parameter set of the projector module.

[1]https://huggingface.co/openai/whisper-large-v3
[2]https://github.com/QwenLM/Qwen2.5

2.3 Simultaneous Speech-to-text Translation

We enable our offline-trained ST system to do simultaneous speech-to-text translation via Local Agreement (LA) (Liu et al., 2020; Polák et al., 2022), which is one of the most commonly used decision policy in recent years. It compares the generated hypotheses of two consecutive chunks and only emit their longest common prefixes (i.e., agreement). A fixed length chunk size (speech segment size) is tuned to control the quality-latency trade-off for SimulST.

2.4 Streaming Speech-to-text Translation

The SimulST system is assumed to work on pre-segmented speech and it is not practical to directly process a long audio stream in real-world scenarios due to latency and computational resources. We develop the StreamST system by integrating an automatic segmenter module into our SimulST system to detect the speech segments $\mathbf{S} = \{s^1, s^2, ...s^N\}$ in real-time. As illustrated in Figure 1, once the segmenter module detect the start point s_1^i of a speech segment s^i, the subsequent modules process the speech chunk-by-chunk in a SimulST manner to emit translations. When the speech segment endpoint is detected, both of the speech and text history buffers are reset, and the translation stops until the start point of the next speech segment is detected.

We use Supervised Hybrid Audio Segmentation (SHAS) (Tsiamas et al., 2022) as the segmentation method for our StreamST system. SHAS is a neural-based method that can effectively learn the optimal segmentation from manually segmented speech corpus to achieve the state-of-the-art segmentation performance. It uses a pre-trained wav2vec 2.0 (Baevski et al., 2020) to extract acoustic features and a SHAS classifier to obtain the probabilities for each audio frame. SHAS determines the speech offset τ and duration Δt of an input audio with a probability threshold θ. However, the SHAS is designed to segment a long audio into multiple speech segments that are shorter than a predefined maximum length L_{max} using the probabilistic Divide-and-Conquer (pDAC) algorithm, while in StreamST, the length of the audio stream increases incrementally.

We enable the SHAS to perform real-time segmentation for StreamST. Specifically, we apply SHAS on the incrementally increasing audio stream until it detects a speech segment offset. The first detected offset is treated as the segment start point, $\mathbf{Seg}_{start}$. Then the subsequent modules of the StreamST system process the speech chunk-by-chunk to generate translations until the segment endpoint $\mathbf{Seg}_{end} = (\tau + \Delta t)$ is detected. However, we observed that SHAS consistently returns an offset-duration pair even when processing incomplete audio streams where speech has not yet finished. In these cases, the SHAS-detected speech segments become too short, negatively impacting the overall performance of the StreamST system. To address this issue, we leverage our empirical observation that when speech is ongoing, the SHAS-detected segment endpoint $\mathbf{Seg}_{end}$ typically falls very close to the length of the currently available audio stream L_{stream}. We therefore introduce a

Algorithm 1 SHAS-based StreamST

Require: Audio stream $\mathbf{X}$, pause length L_{pause}, minimum segment length L_{min}, maximum segment length L_{max}, chunk size $\mathcal{C}$
Ensure: Translation output $\mathbf{Y}$
while processing audio stream **do**
  $\tau, \Delta t \leftarrow$ SHAS($\mathbf{X}$) ▷ Get offset and duration
  if no speech detected **then**
    Continue reading stream
    continue
  end if
  $\mathbf{Seg}_{start} \leftarrow \tau$
  $\mathbf{Seg}_{end} \leftarrow \tau + \Delta t$
  $L_{stream} \leftarrow$ length($\mathbf{X}$)
  segmentComplete $\leftarrow$ False
  if $\mathbf{Seg}_{end} - \mathbf{Seg}_{start} \geq L_{max}$ **then**
    segmentComplete $\leftarrow$ True ▷ Maximum length reached
  else if $\mathbf{Seg}_{end} + L_{pause} < L_{stream}$ **and** $\mathbf{Seg}_{end} - \mathbf{Seg}_{start} > L_{min}$ **then**
    segmentComplete $\leftarrow$ True ▷ Valid pause detected
  end if
  Segment $\leftarrow \mathbf{X}[\mathbf{Seg}_{start} : L_{stream}]$
  if length(Segment) $\geq$ *PrevLength*$+\mathcal{C}$ **then**
    Process segment chunk-by-chunk
    $\mathbf{Y} \leftarrow$ SimulST(Segment)
    PrevLength $\leftarrow$ length(Segment)
  end if
  if segmentComplete **then**
    Reset buffers and prepare for next segment
  end if
end while

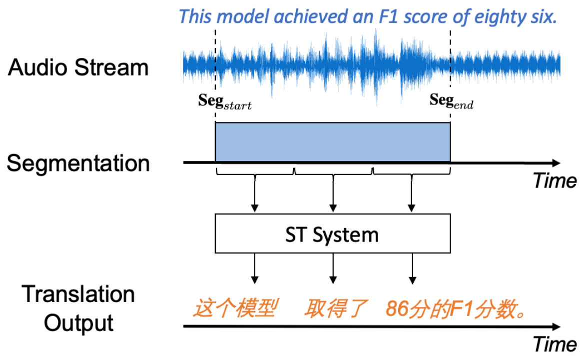


Figure 2: An English-Chinese translation example demonstrating our StreamST system workflow. Upon detecting the speech start point $\mathbf{Seg}_{start}$, the SHAS segmenter triggers the translation system to process incoming speech incrementally, chunk-by-chunk, generating translations continuously until a valid endpoint $\mathbf{Seg}_{end}$ is detected.

pause length parameter L_{pause} and consider a detected segment endpoint $\mathbf{Seg}_{end}$ to be valid only when:

$$\mathbf{Seg}_{end} + L_{pause} < L_{stream}. \tag{6}$$

We demonstrate the significance of the parameter L_{pause} in Section 4.3.3. For practical implementation, we set maximum and minimum segment length constraints to prevent excessively long or short segmentation. Algorithm 1 provides the complete inference procedure for our StreamST system, while Figure 2 illustrates a representative English-Chinese translation example.

3 Experiments Setup

3.1 Data

We used CoVoST-2 (Wang et al., 2020) for all language pairs: English-to-German (En→De), English-to-Japanese (En→Ja), and English-to-Chinese (En→Zh) and also included Europarl-ST (Iranzo-Sánchez et al., 2020) for En→De. We followed our previous submission (Ko et al., 2024) to conduct data filtering based on Bilingual Prefix Alignment (Kano et al., 2022). We used ACL 60/60 (Salesky et al., 2023) data for both validation and evaluation. All of the text data was tokenized using LLM's default tokenizer.

3.2 Evaluation Setup

We assessed the system performance using metrics for both translation quality and latency. For translation quality, we employed BLEU (↑) calculated with SacreBLEU (Post, 2018). For latency evaluation, we used the Length Adaptive Average Lagging (LAAL) (↓) (Papi et al., 2022b) for the SimulST and StreamLAAL (↓) (Papi et al., 2024) for our StreamST system. Additionally, we report the computation-aware versions of both LAAL and StreamLAAL to account for processing overhead. All experiments were conducted using the SimulEval (Ma et al., 2020a) toolkit, providing a standardized evaluation framework.

<System>: You are a professional interpreter who is good at simultaneous interpretation and translation. The user will provide you with a speech in English, which is enlosed within <Speech> and </Speech> tags. And you need to provide both the translation and transcription.

<User>: Based on this original English speech <Speech><SpeechHere></Speech>, complete its translation into <tgt_lang>.

Figure 3: LLM prompt used for both training and evaluation.

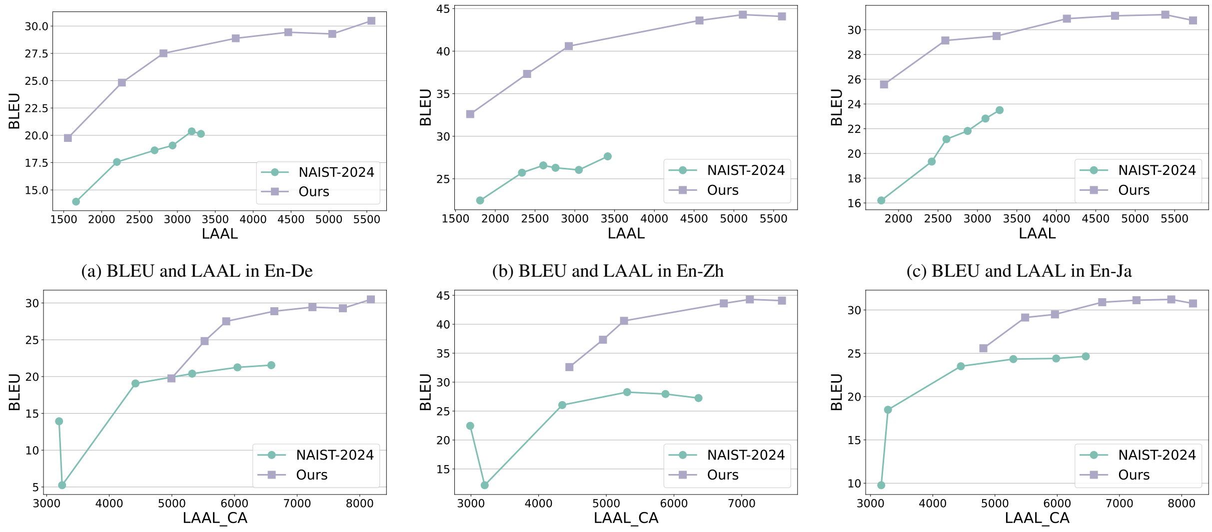
3.3 Offline Model

We trained the model of our system in an offline manner. The speech input was provided as waveforms with 16kHz sampling rate. The Whisper encoder processed this input using a causal attention mask to prevent the model from utilizing future information. The LLM then processed the acoustic embeddings produced by the DeCo projector to generate translations based on a prompt instruction as shown in Figure 3. During training, we used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The learning rate was controlled by a cosine scheduler with a base learning rate of 2.0×10^{-4} and 3,000 warming-up steps within the total 100,000 updates. Validation was performed every 1,500 updates, and model checkpoints were saved based on the best BLEU scores. We averaged the parameters of the ten best-performing checkpoints to create the best model.

3.4 Simultaneous Speech-to-Text Translation

We adapted our offline-trained model for SimulST by applying the local agreement policy to the LLM-based translation system. To control the quality-latency trade-off, we used variable chunk sizes of $\{0.5s, 0.75s, 1.0s, 1.5s, 2.0s, 2.5s, 3.0s\}$. During inference, we employed beam search with a beam size of 4 to generate translation hypotheses for each input chunk.

We compare our SimulST system with our submission from the previous year. The primary dis-



(a) BLEU and LAAL in En-De
(b) BLEU and LAAL in En-Zh
(c) BLEU and LAAL in En-Ja
(d) BLEU and LAAL_CA in En-De
(e) BLEU and LAAL_CA in En-Zh
(f) BLEU and LAAL_CA in En-Ja

Figure 4: Quality-latency trade-off of our **SimulST** system compared to our last year's system on ACL 60/60 dev set.

tinction between the two systems lies in the adoption of an LLM-based model architecture.

3.5 Streaming Speech-to-Text System

We developed our submitted StreamST system by integrating an online version SHAS segmenter with our SimulST model. The pause length L_{pause} and the segmentation threshold θ parameters of SHAS were set differently for each language pair: $\{0.025s, 0.2\}$ for En→De and $\{0.025s, 0.4\}$ for both En→Zh and En→Ja. The impact of these hyperparameters (L_{pause} and θ) is analyzed in Section 4.3.3.

We compare our submitted system with the IWSLT 2025 baseline systems[3]. The baselines implement StreamST using either a naive fixed-length segmenter or a Voice Activity Detection (VAD) segmenter applied to the SeamlessM4T model (Barrault et al., 2023) for all language pairs. An additional cascaded model, which comprises a Whisper ASR model and a M2M100 (Fan et al., 2021) machine translation model, is included for the En→De language pair.

4 Experimental Results

4.1 Offline Results of Topline

The offline performance of our model establishes an upper bound for both the SimulST and StreamST systems by utilizing manual segmentation and processing the complete context to generate translations. Table 1 presents the results of the offline model on the ACL 60/60 dataset.

Table 1: Offline results of our model in the submitted system on ACL 60/60 dev set.

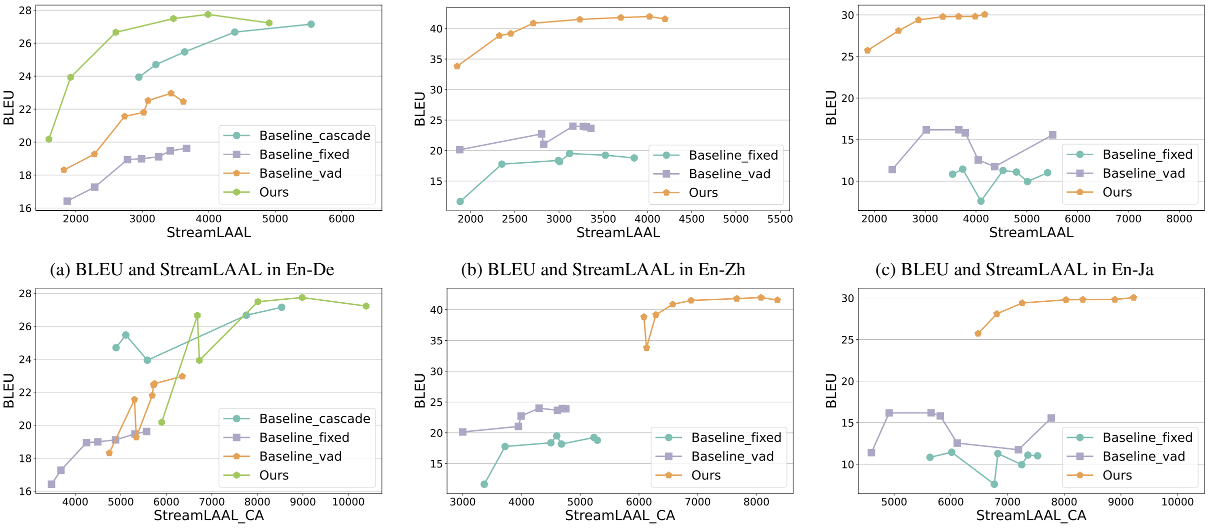
Language Pair	**BLEU Score**
En–De	28.2
En–Zh	43.9
En–Ja	30.3

4.2 Simultaneous Speech-to-text Translation

4.2.1 NAIST 2024 Model vs. 2025 Model

Non-computation-aware latency: We managed to improve our system compared to our system of last year on non-computation-aware latency setting. As can be seen in Figure 4a through Figure 4c, our system outperforms our previous year system by a margin of 6.4 BLEU score on En-De language pair, 12.3 BLEU score on En-Zh language pair, and 5.2 BLEU score on En-Ja language pair when compared at equivalent latency levels. **Computation-aware latency:** We managed to improve our system compared to our system of last year on computation-aware latency setting. As can be seen in Figure 4d through Figure 4f, our current year system managed to improve the overall BLEU score in all pairs of languages with a greater difference in En-Zh translation, as shown by 4e. In computationa-aware setting, our system managed to improve the 6.6 BLEU score on latency

[3] https://github.com/pe-trik/iwslt25-baselines



(a) BLEU and StreamLAAL in En-De
(b) BLEU and StreamLAAL in En-Zh
(c) BLEU and StreamLAAL in En-Ja
(d) BLEU and StreamLAAL_CA in En-De
(e) BLEU and StreamLAAL_CA in En-Zh
(f) BLEU and StreamLAAL_CA in En-Ja

Figure 5: Quality-latency trade-off of our submitted streaming speech-to-text translation (**StreamST**) system compared to IWSLT2025 baseline systems on ACL 60/60 dev set.

Table 2: Results of the submitted streaming speech-to-text translation (**StreamST**) system on ACL 60/60 dev set.

Language Pair	Latency Regime	Chunk Size (s)	BLEU	StreamLAAL (ms)
En–De	Low (0–2s)	0.62	23.92	1921
	High (2–4s)	2.0	27.74	3988
En–Zh	Low (0–2.5s)	0.85	39.17	2455
	High (2.5–4s)	2.0	41.80	3699
En–Ja	Low (0–3.5s)	1.5	29.78	3348
	High (3.5–4s)	2.5	29.81	3982

around 4.35 s and the 12.3 BLEU score on latency around 5.3 s on that particular language pair. Despite not showing as much of a difference, on En-De and En-Ja language pair similar pattern could be observed where our current year system gives better BLEU score overall on similar latency. However, our LLM-based model architecture is more computationally expensive than last year's encoder-decoder model, resulting in higher latency under computation-aware evaluation conditions.

4.3 Submitted StreamST System

In this section, we report the results of our submitted system for IWSLT 2025 simultaneous track. We followed the data condition for both training and evaluation as well as the allowed pre-trained models and therefore our submission is constrained.

4.3.1 Main Results

Figure 5a through Figure 5c illustrate the non-computation-aware quality-latency tradeoff between our StreamST system and the baselines. For the En→De language pair, our system outperforms all three baseline systems in both translation quality and latency metrics, while achieving slightly better peak translation quality compared to the cascaded baseline model. For the En→Zh and En→Ja language pairs, our system also demonstrates substantially superior performance compared to both of the baseline systems.

For each language pair, we select two submission with configurations satisfying the low latency and high latency regimes. Table 2 presents the scores of our submitted StreamST system.

4.3.2 Computation-aware Latency

We also evaluate the computation-aware[4] quality-latency trade-off of our StreamST system, as illustrated in Figures 5d through 5f. While our system demonstrates strong performance under non-computation-aware conditions, it exhibits higher

[4]The computation-aware evaluation was conducted using an NVIDIA RTX A5000 GPU.

latency across all three language pairs when real computation time is considered. This increased latency stems from the LA policy's substantial computational requirements in practical applications. Unfortunately, cross-attention-based decision policies (EDAtt, AlignAtt), which typically perform better under computation-aware conditions, cannot be directly integrated into our LLM-based end-to-end system. This limitation highlights the need to develop more efficient decision policies specifically designed for LLM-based systems in future work.

4.3.3 Ablation Study for SHAS

As mentioned in Section 2.4, we implemented a short pause length to prevent premature segment termination in our SHAS-based StreamST system. To understand the influence of the critical SHAS parameters, we conducted a comprehensive ablation study examining both pause length (L_{pause}) and SHAS threshold (θ). We evaluated offline translation quality across various segmentation configurations with different (L_{pause}, θ) combinations. As shown in Table 3 through Table 5, we identified optimal configurations for each language pairs, $\{0.025s, 0.2\}$ for En→De and $\{0.025s, 0.4\}$ for both En→Zh and En→Ja. Notably, when the pause length parameter L_{pause} was disabled ($L_{pause} = 0.0s$), translation quality decreased significantly across all three language pairs due to premature segment termination. This finding underscores the importance of properly configuring the pause length parameter in SHAS-based segmentation for StreamST systems.

Table 3: Impact of SHAS hyperparameters on En→De.

$\mathbf{L_{pause}}$	**Threshold (θ)**					
	0.6	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
0.0s	14.58	14.85	15.06	15.09	14.53	14.48
0.025s	27.14	28.40	29.82	30.04	**30.85**	30.16
0.05s	27.74	28.80	30.03	30.07	30.78	30.55
0.1s	28.41	28.96	29.06	30.20	30.39	29.38

Table 4: Impact of SHAS hyperParameters on En→Zh.

$\mathbf{L_{pause}}$	**Threshold (θ)**					
	0.6	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
0.0s	33.20	33.85	33.65	33.73	32.84	32.67
0.025s	41.84	42.43	**43.60**	42.03	37.18	34.45
0.05s	41.32	43.09	42.71	41.38	37.40	33.94
0.1s	41.73	42.04	41.40	41.02	36.42	28.98

Table 5: Impact of SHAS hyperParameters on En→Ja.

$\mathbf{L_{pause}}$	**Threshold (θ)**					
	0.6	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
0.0s	25.62	25.74	25.83	25.39	25.27	24.78
0.025s	37.09	37.57	**38.61**	38.27	37.02	36.25
0.05s	37.25	37.45	38.17	38.31	36.74	35.97
0.1s	37.15	37.77	38.19	38.44	36.24	34.95

5 Conclusion

This paper presents our StreamST system developed for the IWSLT 2025 Simultaneous Speech Translation Track. Experimental results demonstrated the effectiveness of employing an large language model (LLM) as the backbone for the speech translation tasks. Our system also showed the effectiveness of applying SHAS segmentation method in real time to handle unbounded audio stream during streaming speech translation. This time, we used the Local Agreement (LA) for our LLM-based system, which results in a higher computational latency in real condition. In the future, we will investigate better decision policy methods for the LLM-based StreamST system.

Acknowledgments

Part of this work is supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP23K21681.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, and 1 others. 2023. Seamlessm4t: Massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

Xi Chen, Songyang Zhang, Qibing Bai, Kai Chen, and Satoshi Nakamura. 2024. LLaST: Improved end-to-end speech translation system leveraged by large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6976–6987, Bangkok, Thailand. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav

Chaudhary, and 1 others. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Biao Fu, Donglei Yu, Minpeng Liao, Chengxi Li, Yidong Chen, Kai Fan, and Xiaodong Shi. 2025. Efficient and adaptive simultaneous speech translation with fully unidirectional architecture. *arXiv preprint arXiv:2504.11809*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Chao-Wei Huang, Hui Lu, Hongyu Gong, Hirofumi Inaguma, Ilia Kulikov, Ruslan Mavlyutov, and Sravya Popuri. 2024. Investigating decoder-only large language models for speech-to-text translation. *arXiv preprint arXiv:2407.03169*.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2022. Simultaneous neural machine translation with prefix alignment. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 22–31, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Yuka Ko, Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Tomoya Yanagita, Kosuke Doi, Mana Makinae, Haotian Tan, Makoto Sakai, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. 2024. NAIST simultaneous speech translation system for IWSLT 2024. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 170–182, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. TransLLaMa: LLM-based simultaneous translation system. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 461–476, Miami, Florida, USA. Association for Computational Linguistics.

Dan Liu, Mengge Du, Xiaoxi Li, Yuchen Hu, and Lirong Dai. 2021. The USTC-NELSLIP systems for simultaneous speech translation task at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 30–38.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Proc. Interspeech*, pages 3620–3624.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587.

Thai-Son Nguyen, Sebastian Stüker, and Alex Waibel. 2021. Super-Human Performance in Online Low-Latency Recognition of Conversational Speech. In *Proc. Interspeech*, pages 1762–1766.

Siqi Ouyang, Xi Xu, and Lei Li. 2025. Infinisst: Simultaneous translation of unbounded speech with large language model. *arXiv preprint arXiv:2503.02969*.

Sara Papi, Marco Gaido, Matteo Negri, and Luisa Bentivogli. 2024. StreamAtt: Direct streaming speech-to-text translation with attention-based audio history selection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3692–3707, Bangkok, Thailand. Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022a. Does simultaneous speech translation need simultaneous models? In *Findings of the Association for Computational Linguistics: EMNLP*, pages 141–153.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022b. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Sara Papi, Matteo Negri, and Marco Turchi. 2023a. Attention as a guide for simultaneous speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

Sara Papi, Marco Turchi, and Matteo Negri. 2023b. AlignAtt: Using Attention-based Audio-Translation

Alignments as a Guide for Simultaneous Speech Translation. In *Proc. INTERSPEECH*, pages 3974–3978.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Simulspeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Haotian Tan and Sakriani Sakti. 2024. Contrastive feedback mechanism for simultaneous speech translation. In *Interspeech 2024*, pages 852–856.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In *Proc. Interspeech 2022*, pages 106–110.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Xi Xu, Siqi Ouyang, Brian Yan, Patrick Fernandes, William Chen, Lei Li, Graham Neubig, and Shinji Watanabe. 2024. CMU's IWSLT 2024 simultaneous speech translation system. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 154–159, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. 2024. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 2461–2474.

Shaolei Zhang, Qingkai Fang, Shoutao Guo, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. StreamSpeech: Simultaneous speech-to-speech translation with multi-task learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8964–8986, Bangkok, Thailand. Association for Computational Linguistics.